**NeurIPS Hide-and-seek Privacy Challenge documentation questionnaire**

**Team name**

|  |
| --- |
| AZWarwickTeam |

**Submission filenames(s)**

|  |  |
| --- | --- |
| Hider | hider\_azwarwick\_submission |
| Seeker |  |

**What class of algorithms does your solution belong to?** (e.g. GANs, VAEs, noise-injection, nearest neighbor, etc.)

|  |  |
| --- | --- |
| Hider | Recurrent AE-WGANs |
| Seeker |  |

**Describe your algorithm in one sentence** (e.g. "Noise is added to the original data and then this data is returned.")

|  |  |
| --- | --- |
| Hider | We train two recurrent networks to translate between time series sample space X and latent space Z, and we train another two networks to act as critic networks and score the fakeness of a given predicted time series. |
| Seeker |  |

**Describe your algorithm in words** (e.g. "Noise is drawn from a Gaussian distribution, with mean 0 and variance s, where the dimension is determined by the size of the dataset. This Noise is added to the original data to produce a noisy version of the dataset and this noisy dataset is then returned as the synthetic data.")

|  |  |
| --- | --- |
| Hider | RAE-WGAN is a four-network model comprising two recurrent WGANs and two recurrent Autoencoders. One WGAN is form by generator G and critic Cx on real samples, the other by generator G and critic Cz on latent space. One autoencoder is trained on real samples G(E(x)) the other is trained on the latent space (noise) E(G(z)). The aim is to obtain a one-to-one mapping (i.e. bijective function) between the latent space Z and the sample space X. After training we expect that WGAN produces realistic samples after transforming the real data G(E(x)). Theoretical analysis imply that our approach may improve the training of GANs by avoiding common pitfalls such as mode collapse and lack of convergence. |
| Seeker |  |

**Specify any loss functions used** (e.g. "No loss functions used.")

|  |  |
| --- | --- |
| Hider | RAE-WGAN requires four-loss functions:   * The samples reconstruction loss ||G(E(x)) – x||\_1 * The latent space reconstruction loss ||E(G(z))-z||\_2 * The Wasserstein loss for samples. Cx is trained to minimize the Wasserstein or Earths Movers distance between actual time series samples (x) and fake time series. * The Wasserstein loss for latent space. Cz is trained to minimize the Wasserstein distance between actual noise (z) and encodings. |
| Seeker |  |

**Specify any hyperparameters and how they are optimized (or preset values)** (e.g. "The noise size, s, is set to 0.1.")

|  |  |
| --- | --- |
| Hider | We have several hyper-parameters for RAE-WGAN. Including the number of critics (5), the coding size (42), batch size (64), learning rate (0.01) and clipping parameter (0.0005). We found that the most interesting hyper-parameter may be the number of critics and the coding size. The coding size refers to the latent features' dimension, that is the size of the encoding of the encoder (and latent space). The hyper-parameter is currently set to roughly the number of features in the dataset (42). It might be possible to outperform the current model by changing the coding size parameter, for example, by learning more useful patterns in the original data. |
| Seeker |  |

**Specify any pre-trained models used by your algorithm** (e.g. "None.")

|  |  |
| --- | --- |
| Hider | We don't use a pre-trained model. However, we have an algorithm for preprocessing the data and conduct imputation of missing values. Missing values are first attempted to be imputed via interpolation. If none values are available for an individual n and variable d then, imputation is initialized with the overall median of the variable d. The initialized imputed data is used to train a Fully connected layer WGAN and then imputed. |
| Seeker |  |

**Pseudo-code for your algorithm**

e.g. **Inputs:** Dataset, D, random seed

**Hyperparameters:** s (default 0.1)

1. Determine dataset dimension: n x d x T

2. Draw N ~ N(0, s), an n x d x T dimensional Gaussian

3. Return D + N

|  |  |
| --- | --- |
| Hider | Inputs: Dataset, d, Mask, m  Preprocess and impute missing values:   1. Determine dataset dimensions: n x d x T 2. Normalize data, obtain min and max vectors of size d 3. Interpolate missing data by n, d for t in sequence along with T 4. Inputs to algorithm Recurrent AE-WGAN = Append imputed data X' and Mask   Start algorithm Recurrent AE-WGAN:  Hyperparameters: code\_size (default 42), n\_critic (default 5), batch\_size (default 64), n\_epoch (default 15) , c the clipping parameter (default 0.01), alpha learning rate (0.0005)   1. While training loss Theta has not converged do: 2. Encoder-Generator optimization on x  * Sample x\_i ~ d a batch of input data * Produce z\_hat by Encoder E E(x) * Reproduce sample with generator x’ <- G(E(x)) * Update EGx using Adam(loss ||G(E(x)) - x||\_1).  1. Generator-Encoder optimization on z  * Sample z ~ Normal(0,1) noise * Produce x\_hat G(z) * Reproduce z’ with E E(G(z)) * Update GEz using Adam(loss ||E(G(z))-z||\_2)  1. Update Cx to critic x\_hat and x’  * for t = 0 … , n\_critic do * Sample x\_i ~ d a batch of input data * Produce x\_hat by encoder E(x) * Reproduce samples by generator x’ <- G(E(x)) * Update Cx RMSPROP(Wassertein(x, x’))  1. Update Cz to critic z\_hat and z’  * for t = 0 … , n\_critic do * Sample z ~ Normal(0,1) a batch of input data * Produce x\_hat by generator G(z) * Reproduce z with E E(G(z)) * Update Cz RMSPROP(Wassertein(z, z’)) |
| Seeker |  |

Finally, alongside this document **please also submit a commented version of your code**. Please include:

- Docstrings for each new class/function defined

- Inline comments for your main function/class

The goal of these comments is to tie the code to the description you have provided here. Please do not alter the actual content of your code - only add comments/docstrings.

**Submitting your documentation and commented code**

Please submit your commented code within a .zip or equivalent file type (1 file per solution), and share it with us as an attachment alongside this Word doc.

You can send these via email (to [nm736@cam.ac.uk](mailto:nm736@cam.ac.uk); [james.jordon@wolfson.ox.ac.uk](mailto:james.jordon@wolfson.ox.ac.uk); [es583@cam.ac.uk](mailto:es583@cam.ac.uk)) or DM James Jordon/Evgeny Saveliev on Slack (you can join the workspace [with this URL](https://join.slack.com/t/hideandseekpr-fbc8582/shared_invite/zt-k2h9xye8-RQNen128uXIG2TRsLa_ppA)).